**DSA GROUP assignment**

**SECTION A: Algorithms representation of the different modules and/functions (Pseudocode/flowchart)**

**[75 marks]**

**Project summary**

A Namibian telecommunications company is looking for an efficient phonebook

algorithm based on basic linear data structures, the key is to use simple, well-known data structures and algorithms that can still deliver reasonably efficient performance for typical phonebook operations (**insert**, **search**, **delete**, **update**). Linear data structures such as arrays and linked lists are relatively simple to implement and understand, making them ideal for simpler systems. Students are free to explore more advanced and efficient structures such as tries, hash tables etcetera.

**A : PSEUDOCODE**

START

MODULE: PhoneBookApplication

FUNCTION main()

DECLARE scanner as Scanner

DECLARE phoneBook as PhoneBook

PRINT "Admin Sign In"

PROMPT "Enter username: "

READ username

PROMPT "Enter password: "

READ password

IF username is "admin" AND password is "ilovedsa" THEN

PRINT "Sign in successfully."

ELSE

PRINT "Invalid credentials. Exiting..."

RETURN

WHILE true DO

PRINT "--- Welcome to Phone Book Menu ---"

PRINT "1. Add Contact"

PRINT "2. Display All Contacts"

PRINT "3. Search Contact by Name"

PRINT "4. Update Contact"

PRINT "5. Delete Contact"

PRINT "6. Exit"

PROMPT "Choose an option: "

READ choice

SWITCH choice DO

CASE 1:

PROMPT "Enter name: "

READ name

PROMPT "Enter mobile number: "

READ mobileNumber

PROMPT "Enter telephone number: "

READ telephoneNumber

PROMPT "Enter email address: "

READ emailAddress

PROMPT "Enter company: "

READ company

PROMPT "Enter department: "

READ department

PROMPT "Enter position: "

READ position

CALL phoneBook.addContact(name, mobileNumber, telephoneNumber, emailAddress, company, department, position)

CASE 2:

CALL phoneBook.displayAllContacts()

CASE 3:

PROMPT "Enter the name to search: "

READ searchName

CALL phoneBook.searchContactByName(searchName)

CASE 4:

PROMPT "Enter the name of the contact to update: "

READ updateName

PROMPT "Enter new mobile number: "

READ newMobileNumber

PROMPT "Enter new telephone number: "

READ newTelephoneNumber

PROMPT "Enter new email address: "

READ newEmailAddress

PROMPT "Enter new company: "

READ newCompany

PROMPT "Enter new department: "

READ newDepartment

PROMPT "Enter new position: "

READ newPosition

CALL phoneBook.updateContact(updateName, newMobileNumber, newTelephoneNumber, newEmailAddress, newCompany, newDepartment, newPosition)

CASE 5:

PROMPT "Enter the name of the contact to delete: "

READ deleteName

CALL phoneBook.deleteContact(deleteName)

CASE 6:

PRINT "Exiting..."

CLOSE scanner

RETURN

DEFAULT:

PRINT "Invalid choice. Please try again."

END SWITCH

END WHILE

END FUNCTION

MODULE: PhoneBook

FUNCTION addContact(name, mobileNumber, telephoneNumber, emailAddress, company, department, position)

DECLARE newContact as Contact

INITIALIZE newContact with parameters

IF head is null THEN

SET head and tail to newContact

ELSE

DECLARE current as Contact

SET current to head

WHILE current is not null AND current.name < newContact.name DO

SET current to current.next

END WHILE

IF current is head THEN

INSERT newContact at the beginning

ELSE IF current is null THEN

INSERT newContact at the end

ELSE

INSERT newContact in the middle

END IF

END IF

PRINT "Contact saved successfully."

END FUNCTION

FUNCTION displayAllContacts()

IF head is null THEN

PRINT "No contacts found."

RETURN

END IF

DECLARE current as Contact

SET current to head

WHILE current is not null DO

CALL current.displayContact()

SET current to current.next

END WHILE

END FUNCTION

FUNCTION searchContactByName(name)

IF head is null THEN

PRINT "No contacts found."

RETURN

END IF

DECLARE current as Contact

SET current to head

WHILE current is not null DO

IF current.name is equal to name THEN

CALL current.displayContact()

RETURN

END IF

SET current to current.next

END WHILE

PRINT "Contact not found."

END FUNCTION

FUNCTION updateContact(name, newMobileNumber, newTelephoneNumber, newEmailAddress, newCompany, newDepartment, newPosition)

DECLARE current as Contact

SET current to head

WHILE current is not null DO

IF current.name is equal to name THEN

UPDATE current with new values

PRINT "Contact updated successfully."

RETURN

END IF

SET current to current.next

END WHILE

PRINT "Contact not found."

END FUNCTION

FUNCTION deleteContact(name)

IF head is null THEN

PRINT "No contacts to delete."

RETURN

END IF

DECLARE current as Contact

SET current to head

WHILE current is not null DO

IF current.name is equal to name THEN

IF current is head THEN

SET head to current.next

IF head is not null THEN

SET head.prev to null

END IF

ELSE IF current is tail THEN

SET tail to current.prev

IF tail is not null THEN

SET tail.next to null

END IF

ELSE

SET current.prev.next to current.next

SET current.next.prev to current.prev

END IF

PRINT "Contact deleted successfully."

RETURN

END IF

SET current to current.next

END WHILE

PRINT "Contact not found."

END FUNCTION

MODULE: Contact

CONSTRUCTOR Contact(name, mobileNumber, telephoneNumber, emailAddress, company, department, position)

SET this.name to name

SET this.mobileNumber to mobileNumber

SET this.telephoneNumber to telephoneNumber

SET this.emailAddress to emailAddress

SET this.company to company

SET this.department to department

SET this.position to position

SET this.prev to null

SET this.next to null

END CONSTRUCTOR

FUNCTION displayContact()

PRINT "Name: " + this.name

PRINT "Mobile Number: " + this.mobileNumber

PRINT "Telephone Number: " + this.telephoneNumber

PRINT "Email Address: " + this.emailAddress

PRINT "Company: " + this.company

PRINT "Department: " + this.department

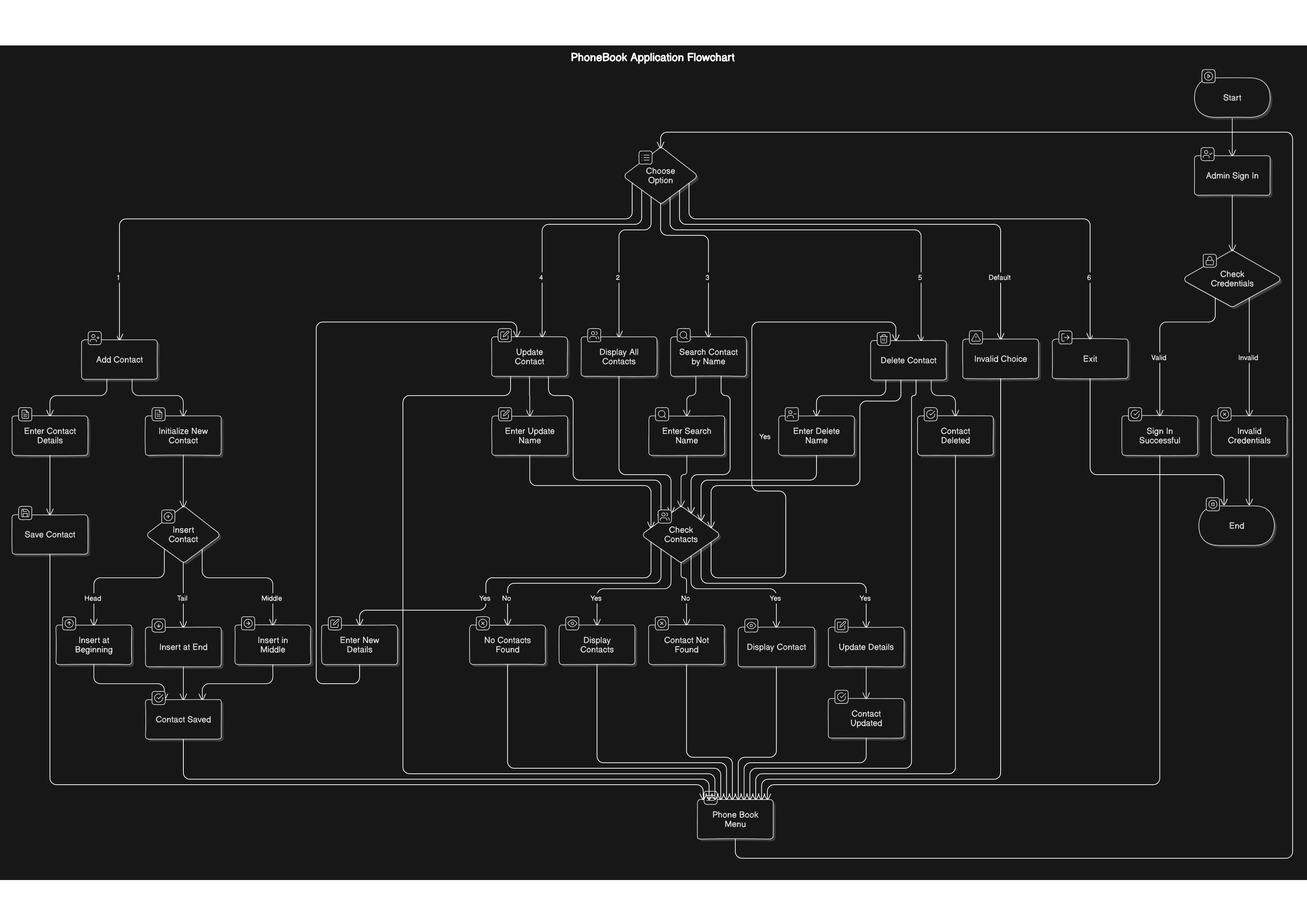
PRINT "Position: " + this.position

PRINT "---------------------------------"

END FUNCTION

END

**B : FLOWCHART**

****

**C: GROUP MEMBERS LOG SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| Student name | Student number | Student specialisation | Contribition(%) |
| Hamutenya Abraham | 224059513 | Computer science | 100 |
| Shipapo John | 224076361 | Informatics | 100 |
| Matuba Immanuel Tetele | 223006068 | Cyber Security | 100 |
| Kuhatumwa Gloria | 224082221 | Computer science | 100 |
| Lisa Ndali Sheehama | 223020281 | Computer Science | 100 |